Katrina Fenlon
Graduate School of Library and Information Science
University of Illinois at Urbana-Champaign
Practicum Final Paper
kfenlon2@illinois.edu
May 2009

Exploring the viability of semi-automated document markup

Digital humanities scholarship has long acknowledged the abundant theoretical advantages of text encoding; more questionable is whether the advantages can, in practice and in general, outweigh the costs of the usually labor-intensive task of encoding. Markup of literary texts has not yet been undertaken on a scale large enough to realize many of its potential applications and benefits. If we can reduce the human labor required to encode texts, libraries and their users can take greater advantage of the hosts of texts being produced by various mass digitization projects, and can focus more attention on implementing tools that use underlying encodings. How far can automation take an encoding effort? And what implications might that have for libraries and their users?

Compelled by such questions, this paper explores the viability of semi-automated text encoding. The immediate context of this paper is an ongoing project[1] to refine a system of automatic transformations capable of transforming the coarse output of a scanning and optical character recognition (OCR) process into a valid TEI[2] document.

1. Related Research

A few decades of theory[3] describe the benefits to be gained by descriptively marking up texts. While encoding schemes have proliferated (trailed by practitioners) there is a notorious divergence between the theory and practice of text encoding. Scifleet et al. (2006), among others, state it frankly:

> There is now a noteworthy echo among researchers that error prone, idiosyncratic practice may turn out to be the hallmark of markup language innovation. Alongside these concerns is an acknowledgement that the functionality achieved through markup languages has been less than was expected and diffusion slower than anticipated. -Scifleet et al, 2006

The humanities encoding projects that have managed to take root generally deal with niche content – sub-specializing within the mostly literary domain of the TEI[4] – and are often based on not-readily-interoperable customizations of the TEI schema (whether those customizations are made consciously and documented, or emerge as workarounds to encoding

[1] Spearheaded by Timothy Cole at the University of Illinois at Urbana Champaign and Martin Mueller at Northwestern University
[2] Text Encoding Initiative (in this case, "TEI" is standing for the TEI Guidelines for…). See TEI SITE and Ide and Sperberg-McQueen, 1995.
[3] Let's say the tradition begins with Goldfarb (1981). For debates about the nature and usefulness of descriptive markup see e.g. Coombs, Renear and Derose (1987) and Caton (2000)
[4] The list of TEI projects maintained by the Initiative demonstrates this: http://www.tei-c.org/Activities/Projects/. Note that TEI is not the only encoding standard, but is by far the major standard in use for literary texts.

challenges). Most existing projects rely heavily or completely on manual encoding. They are in effect boutique projects, appealing to small communities of interest.

In contrast, many university libraries and content providers are involved in mass digitization and are facing the question of what to do with material once it has been digitized. Reflecting this trend, the theme of the 2009 TEI Annual Meeting is, "Text encoding in the era of mass digitization".[5] Particularly relevant to us, the University of Illinois Library, partnering with the Open Content Alliance (OCA), is digitizing a collection of 19th-century British novels; part of this project's objective is to determine whether the resulting texts can be efficiently encoded, such that they can cooperate with existing TEI-encoded collections, e.g. the vast Text Creation Partnership[6] collections, which will soon enter the public domain.

Some research has gone into marking up and using large corpora. Besides TCP and the Women Writers Project,[7] which are standout initiatives of girth, Project Gutenberg's Distributed Proofreaders[8] project has broached the problem of mass OCR-error correction in plain text documents by drawing on the power of distributed (volunteer) human computing. The DP model could eventually be useful for the correction of both OCR and markup errors, should large-scale automated encoding become a reality. Projects that focus on encoding for the sake of linguistic analysis perforce deal with large collections: the MONK[9] project, WordHoard,[10] and others frequently use some sort of automation (e.g. MorphAdorner[11] software) for word-level markup, using automatically generated metadata to make centuries' worth of texts comparable. Indeed, much of the previous work on automated encoding comes from this realm of information extraction.[12]

2. Our Findings

Our system of automatic transformations was designed for several 19th-c. British novels from the University of Illinois library collection, which were scanned by the OCA and are now available publicly in various formats through the Internet Archive.[13] Our transformation begins with an XML version of the DjVu page image format (filename ending in "djvu.xml") – a product of the OCR process – which contains detailed presentational markup derived from the geography of the scanned page. Importantly, djvu.xml identifies pages, paragraphs, lines, words and their position coordinates (see Fig. 1). Whatever particular OCR software a digitization effort employs, if that software can produce an XML-based format containing accurate, similarly fine-grained markup, then the resulting texts should be amenable to some variation of our workflow – meaning that our workflow could be generalized. A series of five XSLT stylesheets, two of which require human beings to define certain parameters, convert the presentational DjVu

[5] "2009 Conference and Members' Meeting of the TEI Consortium : Call for Proposals"
[6] Text Creation Partnership at the University of Michigan: http://www.lib.umich.edu/tcp/
[7] WWP homepage: http://www.wwp.brown.edu/
[8] Distributed Proofreaders homepage: http://www.pgdp.net/c/
[9] Metadata Offer New Knowledge: http://monkproject.org/
[10] WoardHoard homepage: http://wordhoard.northwestern.edu/userman/index.html
[11] MorphAdorner homepage at Northwestern University: http://morphadorner.northwestern.edu/
[12] See e.g. Abolhassani et al. (2003) and Taghva et al. (1999); Taghva et al. (1996) discusses automatic removal of OCR "garbage strings"
[13] For example, most of our testing was done on Dickens's *Bleak House*, which is found here: http://www.archive.org/details/bleakhouse00dickrich, with an index of available derivatives here: http://ia341009.us.archive.org/2/items/bleakhouse00dickrich/

markup into descriptive TEI markup.[14] The resultant TEI document features bibliographic metadata – derived from a MARC record that accompanies each scanned text – and structural markup of the body of the text, including chapter divisions (with chapter heads), paragraph divisions (paragraphs spanning page breaks are merged), and page and line breaks (see Fig. 2). The appendix contains details on the stylesheets and processing, including a brief description of each stage of the transformation and what information it requires of human encoders.

```
</PARAGRAPH>
<PARAGRAPH>
<LINE>
<WORD coords="893,1317,952,1283">IN</WORD>
<WORD coords="993,1317,1287,1282">CHANCERY.</WORD>
</LINE>
</PARAGRAPH>
<PARAGRAPH>
<LINE>
<WORD coords="200,1442,486,1395">LONDON,</WORD>
<WORD coords="558,1440,854,1395">]\Iicliat&apos;lmas</WORD>
<WORD coords="882,1440,1015,1395">Term</WORD>
<WORD coords="1055,1450,1192,1395">lately</WORD>
<WORD coords="1217,1448,1335,1410">over,</WORD>
<WORD coords="1377,1438,1467,1393">and</WORD>
<WORD coords="1503,1437,1579,1393">tlie</WORD>
<WORD coords="1621,1437,1744,1391">Lord</WORD>
<WORD coords="1779,1437,2044,1390">Chancello]&apos;</WORD>
</LINE>
<LINE>
<WORD coords="136,1522,298,1462">sitting</WORD>
<WORD coords="336,1507,385,1462">in</WORD>
<WORD coords="431,1507,662,1461">Lincoln&apos;s</WORD>
<WORD coords="710,1505,798,1460">Lni</WORD>
<WORD coords="845,1506,968,1460">Hall.</WORD>
<WORD coords="1053,1521,1328,1460">Implacable</WORD>
<WORD coords="1374,1504,1634,1459">November</WORD>
<WORD coords="1679,1503,1888,1458">weatlier.</WORD>
<WORD coords="1975,1501,2041,1457">As</WORD>
</LINE>
```

Fig. 1. The chapter heading and first two lines of the first paragraph of Dickens' *Bleak House*, marked up in djvu.xml

```
<body>
   <div xml:id="div1">
      <head> CHAPTER I. </head>
      <p xml:id="para1106">
         <lb n="11.02"/>IN CHANCERY. </p>
      <p xml:id="para1108">
         <lb n="11.03"/>LONDON, ]\Iicliat'lmas Term lately over, and tlie Lord Chancello]'
         <lb n="11.04"/>sitting in Lincoln's Lni Hall. Implacable November weatlier. As
         <lb n="11.05"/>mncli mnd in the streets, as if the waters liad but newly retired from the
         <lb n="11.06"/>face of the earth, and it would not be wonderful to meet a ]\reg-alosaurns,
         <lb n="11.07"/>forty feet long or so, waddling like an elephantine lizard up IIol])orn-hi]1.
         <lb n="11.08"/>Smoke lowering down from chimuey-]Dots, making a soft black drizzle,
         <lb n="11.09"/>with flakes of soot in it as big as fidl-grown snow-flakes – gone into
         <lb n="11.10"/>mourning, one might imagine, for
```

Fig. 2. The start of *Bleak House* after semi-automatic transformation from djvu.xml into TEI

[14] We rely on a schema called TEI-Analytics, which is like TEI-Lite (P5) augmented for linguistic analysis. The differences between these two schemas have no bearing on the automatic encoding, which focuses on the structure of the documents rather than word-level markup; TEI or TEI-Lite could easily be substituted for TEI-A.

A complete transformation from downloaded djvu.xml document to final TEI-A, using our stylesheets, currently takes an individual encoder approximately twenty minutes. The time investment and the risk of error, which already represent substantial improvements over manual encoding, could be reduced by adding heuristics for determining the values of certain parameters automatically.[15] The transformations are error-prone for a couple of reasons. First, automatically 'upconverting' from presentational to descriptive markup means relying on deductions that can be made based on the features of a text that a computer can recognize, e.g. whitespace, and texts handle these features idiosyncratically. Second, widely varying OCR quality can cause incorrect tagging or even the loss of whole paragraphs of text. To minimize the risk of this and other errors, the workflow relies on an error-seeking stylesheet, detailed in the appendix, which, when run after any phase of the transformation, can sift out missing divisions (including paragraphs and page breaks) and other indicators of textual loss. It is not a perfectly robust solution, and can only indirectly find 'lossless' tagging errors (which can be very difficult for a person to spot, too, in a large, valid text), but it reduces the risk of errors that result in textual loss to the point of nonoccurrence.

In summary, we have established a workflow capable of transforming an XML derivative of scanning/OCR into a TEI-conformant, structurally tagged document, with low human involvement and acceptable error levels. The transformations have been tested on 19th-c. British novels that result from a specific OCR process, but could be adapted to other genres of text and limited to other kinds of OCR output. What remains to be done: Besides stylesheet-specific improvements that are noted in the appendix, the workflow as a whole desires refinement and should be tested on a larger sample of novels. We plan to test it on multi-volume works to evaluate how the presumably fewer differences between related texts affect the efficiency of encoding. The workflow would also benefit from a stronger error-finding method, and we would like to determine to what extent we can automatically encode novels' complex front- and back-matter. More ideally, the system of transformations would be manageable by novice encoders through a user-friendly interface. Most ideally, transformations would be fully automatic, requiring no human-contributed parameters or error-checking. While this appears impossible now, we can certainly get much closer to that ideal than we currently stand, perhaps by relying on deductive algorithms for determining the values of parameters.

3. Implications for Libraries

The concerns expressed by Scifleet et al. (above) – including the idiosyncrasy and sluggish dissemination of markup practice, and the disappointing functionality gained by markup – could be mitigated by automating or semi-automating even the early stages of text encoding. A sufficiently robust workflow could promote sharing (of workflow, of schemas, and of resulting texts) among projects and libraries, thereby helping disseminate both the practice and the rich, interoperable texts. Automated encodings, while more prone to errors rooted in semantics, should however contain fewer idiosyncratic errors than manual encodings (i.e. errors will not vary by encoder), which should therefore be easier to find and fix en masse. Establishing a large corpus of works would encourage the development of applications that tap the increased functionality offered by encoding: a substantial, diverse collection could garner a larger user community to drive demand for applications, and computer-driven literary or linguistic analyses

[15] The appendix notes places where each stylesheet may admit improvement.

would improve with increased samples. In the realm of boutique encoding projects, automatic transformations could provide a customizable basis for subsequent rich encoding and proofreading.

The thrust is this: automated encoding can improve a library's ability to add value to digital collections. The relevance of libraries depends in part on how they can augment resources – like digital versions of public domain works – that are by no means scarce. The efficiency of auto-encoding offers impetus and means for the development of collections and tools that help scholars and readers do what they need to do with unprecedented ease and power. There are of course pragmatic concerns that must ground our thinking about auto-encoding. Though fairly efficient once in place, the establishment or customization of a system of automatic transformations requires a significant investment of expert labor and collaboration. A collection of encoded texts will require more complex content management and information retrieval support than other texts. Because encoding opens new doors for interactivity between users and works, e.g. through annotation and tagging systems, increased attention will have to be given to maintaining the authenticity and integrity of digital texts – to things like auto-generated provenance metadata and databases with sturdy versioning facilities. Preservation of encoded texts will also demand innovations from librarians, though the fact that text encoding relies on open-source, well-documented standards should simplify that problem. The results of our attempt to develop an automated transformation are auspicious enough that these implementation problems are worth considering, at least tentatively, and that debates about the worth of text encoding can focus more on prospective applications than on the feasibility of marking up texts in the first place.

Appendix: Documenting the Stylesheet Transformations

Outline of workflow:

Stage #	Stylesheet Description	Stylesheet Name
1	Transform MARC file associated with text to TEI Header Fragment	
2	Initial djvu.xml to TEI transform	djvu2tei.xsl
3	Add div elements around chapters	chapterDiv.xsl
4	Merge paragraphs that span page breaks	modifiedMerge.xsl
5	Remove seg elements from within paragraphs	noseg.xsl
*	Seek errors (especially missing elements) after any stage of transform	errorSeeker.xsl

Details of each stage:

Stage #	1
Approximate customization time	0
Input	MARC file (from OCA)
Output	TEI Header fragment file (*.xml)
Required customizations (parameter names)	None that I am aware of
Risks	None that I am aware of
To-Do	Nothing that I am aware of

Stage #	2
Approximate customization time	5-15 minutes
Input	*djvu.xml and TEI Header fragment file
Output	*stage1.xml
Required customizations (parameter names)	• (pbAlign) Determine difference between book and page-image pagination, so that pb numbers can be aligned • (firstBodyPage) Enter book page number of first page containing body text • (lastBodyPage) Enter book page number of last page containing body text • (ignoreAboveOdd) Enter approximate y coordinate above which text on an odd-numbered page should be considered a running header (will ignore paragraphs where 2nd coord

	of first word of first line of paragraph is less than or equal to ignoreAbove value (i.e., to suppress running head), could default to typical header margin) • (ignoreAboveEven) Enter approximate y coordinate above which text on an even-numbered page should be considered a running header (will ignore paragraphs where 2nd coord of first word of first line of paragraph is less than or equal to ignoreAbove value (i.e., to suppress running head), could default to typical header margin) • (ignoreBelowOdd) Enter approximate y coordinate below which text on an odd-numbered page should be considered a running foot (will ignore paragraphs where 4th coord of first word of first line of paragraph is greater than or equal to ignoreBelow value (i.e., to suppress running footer), could default to page-height attribute value minus typical footer margin) • (ignoreBelowEven) Enter approximate y coordinate below which text on an even-numbered page should be considered a running foot (will ignore paragraphs where 4th coord of first word of first line of paragraph is greater than or equal to ignoreBelow value (i.e., to suppress running footer), could default to page-height attribute value minus typical footer margin) • (chapterDiv) This parameter is useless if using stage 3 transformation (in which case, use value “False”). Enter "True" or "False" depending on whether chapters in text are designated by string "CHAPTER" • (TEIHeaderFragFileName) Enter name of file containing TEI Header fragment from stage 1
Risks	• If the values of any parameters are inaccurate, text could be removed from output
To-Do	• Consider how to treat front and back matter (as it stands, these are removed from text) • Remove superfluous chapterDiv param • Figure out how to determine the ignoreAbove… and ignoreBelow… params automatically, or default them to appropriate values

Stage #	3
Approximate customization time	5-15 minutes (depending on whether chapter titles need to be manually corrected so that they are uniform/can be caught with regular expression)
Input	*stage1.xml
Output	*chapterDiv.xml
Required customizations	• (chapterHeadRegex) Enter a regular expression to capture

(parameter names)	all and only chapter heads – div elements will be based on this (usually string "CHAP" works, since most texts use capitalized "CHAPTER" before chapter titles) • (runningHeadOption) Enter "True" if you would like to use regular expressions to find and remove any remaining running heads in the document • (runningHeadLeft and runningHeadRight) Enter regular expressions to catch running heads on odd and even pages
Risks	• If parameter values are inaccurate, entire chapters could be mis-tagged or text could be mistaken for running head and deleted; these errors are however easily identifiable
To-Do	• Hard-code or parametrize namespaces (right now relying on '*' XPath trick) • Determine how stylesheet could be made more efficient • Determine where big blocks of whitespace following divs in output are coming from

Stage #	4
Approximate customization time	0
Input	*chapterDiv.xml
Output	*merged.xml
Required customizations (parameter names)	None that I am aware of
Risks	• Problems with punctuation logic (set is limited); if there is a failure, b/c of missing or misinterpreted punctuation, to recognize that a paragraph should or should not merge, that paragraph could be lost, as well as intervening pb element • If not all running heads/feet are removed prior to this stage, they will be merged in place of correct body paragraphs
To-Do	• Improve the logic to reduce aforementioned risks

Stage #	5
Approximate customization time	0
Input	*merged.xml
Output	*noseg.xml
Required customizations (parameter names)	None that I am aware of
Risks	None that I am aware of

To-Do	Nothing that I am aware of

Stage #	* (run after any transformation)
Approximate customization time	0
Input	*.xml
Output	error report text file
Required customizations (parameter names)	None
Risks	• Results should not be interpreted as comprehensive – may not catch all errors in the document (only displays total number of pages, chapters, and paragraphs, and lists missing pages and paragraphs, based on their attribute numbers
To-Do	• Tidy up the stylesheet and output • Make error-seeking logic more robust

References

2009 Conference and Members' Meeting of the TEI Consortium : Call for Proposals. (n.d.). . Retrieved April 30, 2009, from http://www.lib.umich.edu/spo/teimeeting09/proposals.html.

Abolhassani, M., Fuhr, N., & Govert, Norbert. (2003). Information extraction and automatic markup for XML documents. In Lecture Notes in Computer Science. Springer. Retrieved February 19, 2009, from http://www.is.informatik.uni-duisburg.de/bib/pdf/ir/Abolhassani_etal:03.pdf.

Caton, P. (2000). Markup's current imbalance. Markup Languages, 3(1), 1-13. doi: 10.1162/109966201753537123.

Distributed Proofreaders. (n.d.). . Retrieved May 6, 2009, from http://www.pgdp.net/c/.

Goldfarb, C. (1981). A generalized approach to document markup (pp. 68-73). Portland, OR: ACM. doi: 10.1145/800209.806456.

Ide, N., & Sperberg-McQueen, C. (1995). The TEI: History, goals and future. Language Resources and Evaluation, 29(1), 5-15. doi: 10.1007/BF01830313.

MorphAdorner. (n.d.). . Retrieved May 6, 2009, from http://morphadorner.northwestern.edu/.

Mueller, M. (n.d.). TEI Members Meeting 2008: TEI-Analytics and the MONK Project. Retrieved May 6, 2009, from http://www.cch.kcl.ac.uk/cocoon/tei2008/programme/abstracts/abstract-169.html.

Northwestern University. (2009, April 1). WordHoard - Title Page & Table of Contents. Retrieved May 6, 2009, from http://wordhoard.northwestern.edu/userman/index.html.

Palowitch, C., & Stewart, D. (1995). Automating the Structural Markup Process in the Conversion of Print Documents to Electronic Text. Austin, TX. Retrieved February 19, 2009, from http://www.csdl.tamu.edu/DL95/papers/palowitc/palowitc.html.

Renear, A., Dubin, D., & Sperberg-McQueen, C. (2002). Towards a semantics for XML markup (pp. 119-126). McLean, VA. doi: 10.1145/585058.585081.

Taghva, K., Condit, A., & Borsack, J. (1995). An Evaluation of an Automatic Markup System. In Proc. 10th Intl. Conf. on Systems Engineering. San Jose, CA. Retrieved May 6, 2009, from http://www.isri.unlv.edu/publications/isripub/Taghva95a.ps.

Taghva, K., Condit, A., & Borsack, J. (2001). Automatic Removal of "Garbage Strings" in OCR Text: An Implementation . Orlando, FL. Retrieved May 6, 2009, from http://www.isri.unlv.edu/publications/isripub/Taghva01d.pdf.

TEI: Projects Using the TEI. (n.d.). Retrieved April 30, 2009, from http://www.tei-c.org/Activities/Projects/.

University of Michigan Library. (n.d.). Text Creation Partnership. Retrieved April 30, 2009, from http://www.lib.umich.edu/tcp/.

WWP. (n.d.). The Brown University Women Writers Project. Retrieved May 6, 2009, from http://www.wwp.brown.edu/.